calntegrator Install Guide

19 January 2006

Table of Contents

Introduction to caIntegrator 3
How to use the installation guide 3
General Requirements 4
Targeted Developer 4
General Hardware Requirements 5
Obtaining the caIntegrator Source 6
Database Technology 8
Assumptions and Requirements 8
Initialization 8
Verification 10
Analysis Server 11
Assumptions and Requirements 11
Installation 12
Running the Analysis Server 13
Troubleshooting 14
Rembrandt Application 16
Assumptions and Requirements 16
Installation/Deployment 17
Verification 20
Known Issues 21
WebGenome Integration 22
Assumptions and Requirements 22
Installation 24
Troubleshooting 25
Verification 26
Deployment Recommendations 26
WGI Services 26
Appendix A: caIntegrator-spec 28
Assumptions and Requirements 29
Initialization 29
Appendix B: caIntegrator-analysis-commons 30
Assumptions and Requirements 30
Initialization 31
Appendix C: caIntegrator Downloads 31
Documentation 31
Source Distribution 32
Database Download 32
Appendix D: 3rd Party Tools 32

Section 1

Introduction to caIntegrator

"Informatics tools to integrate clinical trials and high throughput molecular analysis"

The caIntegrator knowledge framework provides researchers with the ability to perform ad hoc querying and reporting across multiple domains. This application framework comprises an n-tier service oriented architecture that allows pluggable web-based graphical user interfaces, a business object layer, server components that process the queries and result sets, a data access layer and a robust data warehouse.

caIntegrator is envisioned to be the foundation for a number of translational applications. One such reference implementation at NCICB is called Rembrandt (**Re**pository of **M**olecular **BRA**in **N**eoplasia **D**a**T**a) – http://rembrandt.nci.nih.gov. This knowledge framework offers a paradigm for rapid sharing of information and accelerates the process of analyzing results from various biomedical studies with the ultimate goal to rapidly change routine patient care.

More information is available at the caIntegrator website: http://caintegrator.nci.nih.gov.

How to use the installation guide

The primary components of caIntegrator which will be elaborated upon include: the database, the caIntegrator-spec module, the Analysis Server, the WebGenomeIntegration Module, and the Rembrandt Web Application. The guide will include specific sequential instructions on how to obtain, install, configure, and manage each component.

Throughout the guide, specific elements will be highlighted with a unique style, in order to emphasize and categorize these elements.

Please refer to the key below for definitions of the elements and their corresponding style:

Element	Style
Links	URL's and paths will be styled as blue underlined
Commands	OS console commands will be styled as **courier new bold red**
Messages	expected messages during installation **bold green**
Filenames	file names will by styled in ***bold italics***
Properties/Variables	property and variable names will be styled in **bold orange**
`Code`	code or XML (typically for configuration) will be styled as `courier new bold gray`

General Requirements

The general requirements common to multiple components of caIntegrator are a prerequisite, and must be installed on your machine prior to attempting to work with the caIntegrator framework. The following are open source technologies which power caIntegrator:

- Java Software Development Kit (JDK) version 1.5.0_04
  http://java.sun.com/j2se/1.5.0/download.jsp
- EJB Container (recommended: Jboss version 4.0.2)
  http://www.jboss.com/products/jbossas/downloads
- Jakarta Ant version 1.6.2
  http://archive.apache.org/dist/ant/binaries/
- Oracle 9i
  http://www.oracle.com

Please acquire each of these and follow the installation instructions provided with each respective product for your environment.

Targeted Developer

Although we have tried to make every effort to simply these instructions as much as possible, we feel that this installation guide is best suited for an experienced java developer who is familiar with the following J2EE technologies:

- Unix/Linux environment

(Configuring environment variables; Installing Ant, JDK, and JBOSS server)

- Ant build scripts
- J2EE web application development using the Struts framework, Servlet/JSP's, javascript, AJAX, and XML/XSLT.
- J2EE middle-ware technologies such as n-tier service oriented architecture and software design patterns.
- JBossMQ and Java Messaging Service (JMS)

- Enterprise Java Beans (EJB)
- Installation and execution of R

In addition, you will need assistance / access from an Oracle 9i database administrator to properly configure the database.

General Hardware Requirements

caIntegrator has been tested with the following configuration, as 3 separate machines.

Analysis Server:

The hardware currently supporting the analysis server for the Rembrandt application consists of a cluster of three HP Proliant DL140 G2 machines, with dual 3.60 GHz Xeon 64T cpus, 4 GB ram, mirrored 36 GB SCSI hard drives running the Red Hat EL 3.0 operating system.

The analysis computations can take a significant amount of memory and therefore it is recommended that the machine(s) running the analysis server have at least 2 GB of memory. In addition, the analysis server is intended to be run on a Unix based operating system.

If the analysis server is to be hosted on the same machine that is hosting the Rembrandt application then is recommended that machine have more than one cpu and have at least 4 GB ram (i.e., 2GB dedicated to the analysis server and 2GB dedicated to the application).

Application Server:

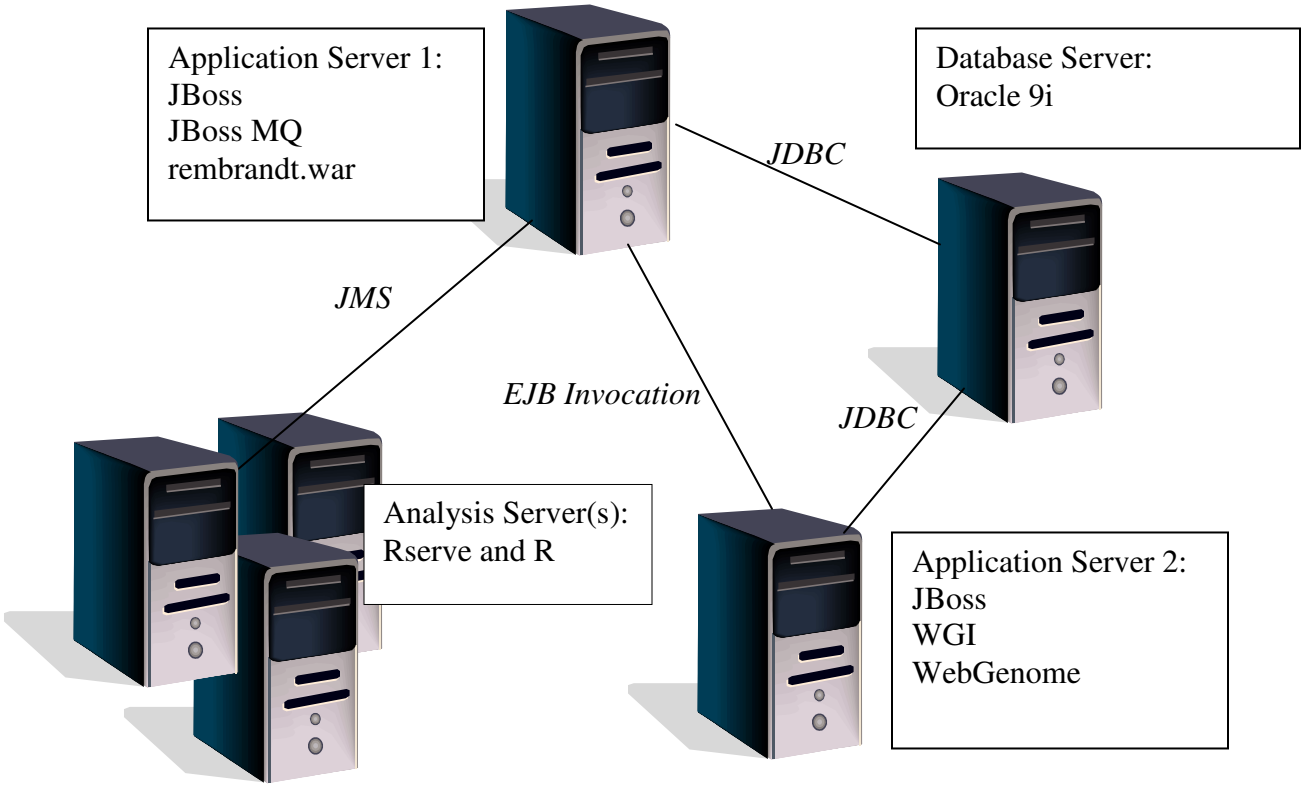
Red Hat EL 3.0 on HP Proliant DL360 G4, with dual 3.40 GHz Xeon 64T cpus, 8 GB ram, mirrored 144 GB SCSI hard drives.

Database Server:

Sun Solaris 8.0 on Sun v1280, with four 900 MHz US-III cpus, 8 GB ram, mirrored local and SAN disks.

Server Configuration

We recommend the following configuration:



Obtaining the caIntegrator Source

The caIntegrator source bundle will be distributed through the NCICB Download center, or via the caIntegrator website http://caintegrator.nci.nih.gov/caintegrator/downloads. All distributions will be in the form of zip archives, and will require a zip utility such as

WinZip, the Java Jar utility, or 7-Zip to open. Begin by selecting the bundled download named "*caIntegrator1.0_src_bundle.zip*" pictured below.

Source Distribution
The source code distribution below contains the caIntegrator source and its reference implementation (REMBRANDT).
Source Code *caIntegrator1.0_src_bundle.zip*

Save this archive, *caIntegrator1.0_src_bundle.zip*, to your machine and extract it into a local directory. After this extraction, you will see 5 main components:

- *webGenomeIntegration.zip*
- *caintegrator-analysis-commons.zip*
- *caintegrator-analysis-server.zip*
- *caintegrator-spec.zip*
- *rembrandt.zip*

Each of these archives will be referenced in the subsequent portions of this guide. Continue by downloading the database initialization files, pictured below, named *rembrandt_db_instructions.zip*. This archive will contain the required files for properly setting up the database for the REMBRANDT reference implementation of caIntegrator.

Database Download
The file below contains all necessary instructions and information for importing the database file for the REMBRANDT reference implementation.
Database Instructions *rembrandt_db_instructions.zip*

Section 2

Database Technology

caIntegrator employs a basic star schema with modification for the Study data warehouse design that supports the integration of clinical and genomic data. It is a generic, query optimized schema that contains fact tables such as "Differential_Gene_Expression_Fact" and "Genomic_Abnormality_Fact", etc. Look up entities such as Genes, Biosample, and Disease type make up the dimensions in the schema. This schema provides a highly de-normalized view of the data and a data neutral framework from which queries can be executed with quick retrieval time.

Assumptions and Requirements

Oracle database 9i has been set up and you have access to the administrator account. You have proper privileges to create a named tablespace and you are familiar with the oracle DB technology.

In addition to the above assumptions, it is also required that you have properly downloaded the *rembrandt_db_instructions.zip* file from the NCICB Download Center as previously directed.

Initialization

Unzip the *rembrandt_db_instructions.zip* to a directory on your local machine, such as C:\caintegrator\db.

Login into Oracle as database the administrator

Create a named tablespace with at least 8 GB in size.

Create a user with this tablespace as default tablespace.

At this point, you have 2 options: importing a dump which will set up the database with the distributed data, or running the DDL, which will set up the DB structure, but will not import any data.

OPTION 1
Create database by importing the DMP file from the download site.
Locate the file *remb.dmp* from the extracted location (labeled below as <location>)
Issue the following command substituting the <variables> with your settings:

```
Imp <username>/<password>@<database>
file=<location>/remb.dmp log= <name.log> full=y
```

OPTION 1 Example:
1) Extract *rembrandt_db_instructions.zip* file to your local C drive. Verify that the extracted file c:\remb.dmp exists.
2) Go to the command line to login to the database named 'rembdb' as the user named 'remb' with password 'rembpass', which the database administrator created for this import. Then issue the following command:
Prompt>**imp remb/rembpass@rembdb file=c:\remb.dmp log=remb.log full=y**

Hit <ENTER>

Note: You/DBA can verify this database import by checking the log file for errors.

OPTION 2
Create a database by running the DDL script from the download site.
Locate the file *remb_ddl.sql* from the extracted location (labeled below as <location>)
Log into the database as created named user
Issue the following command:

```
@<location>/remb_ddl.sql
```

OPTION 2 Example:
1) Extract *rembrandt_db_instructions.zip* to your local C drive. Verify that the extracted file c:\remb_ddl.sql exists.
2) Log into the database named 'rembdb' as the user named 'remb' with password 'rembpass', through SQL Plus, then issue the following SQL statement:

```
@c:\remb_ddl.sql
```

Hit <ENTER>

Verification

Once the database has been created through importing (option1) or the database has been created through running the DDL script (option 2) without error, it can be verified by comparing the number of the objects in current database with the number of the objects from the original database. This verification process can be performed through a SQL query in the SQL Plus or TOAD environment. For both options, the number of objects should match the list below taken from the original Database

Tables 92
Views 11
Triggers 7
Indexes 179
Sequences 12

In addition, most of the tables in the option 1 database should contain seed data for the public to view, and no data is available in option 2 database.

Section 3

Analysis Server

The Analysis Server performs on-the-fly statistical calculations for the caIntegartor application. It can be hosted on the same machine that runs the Rembrandt application or it can be run on any number of remote compute machines. In the remote configuration, each compute machine communicates with the Rembrandt application via the JBossMQ Java Messaging Service (JMS) provided by JBoss. It is important to note that each compute machine does not require its own JBoss instance. The Analysis Server relies on the JMS queues defined in the JBoss instance running the caIntegator application, such as Rembrandt.

This installation was developed and tested on Linux RedHat 3.0.

If configuring multiple compute machines, step 2 only needs to be performed once since all of the compute machines rely on the JBoss instance hosting the application. Steps 1, 3, and 4 should be performed on each compute machine.

Assumptions and Requirements

The following assumptions are in place:
You have properly downloaded the caintegrator bundle, and located the nested archive, *caintegrator-analysis-server.zip*. This archive contains the data, code, and necessary libraries to build and run the analysis server.

You have Java and Ant installed, as listed in the general requirements.

The bundle contains the caintegrator-analysis-commons pre-built jar, as listed in the table below. This jar can also be built from source if desired.

Software	Version	Included in the bundled caIntegrator 1.0 D/L
caintegrator-analysis-commons	1.0	Yes

R and Rserve have been installed. Please see the Rserve homepage at:
http://stats.math.uni-augsburg.de/Rserve/

And the Rserve installation instructions:
http://stats.math.uni-augsburg.de/Rserve/doc.shtml

It is recommended that you read the Rserve installation instructions before installing R. Specifically, the Rserve instructions recommend building the R distribution from source using the **`--enable-R-shlib`** flag.

The R source distribution can be obtained from R project site:
http://www.r-project.org/ (click the link under download)

Installation

Follow the sequence to properly install the Analysis Server.

1) Unzip the file *caintegrator-analysis-server.zip*
On the machine(s) that will be used to run the analysis server, extract the file *caintegrator-analysis-server.zip* to the directory where the software is to be installed. The subdirectory named caintegrator-analysis-server which is located under the caintegrator-ver1.0 directory will be referred to as the "analysis server install directory" in subsequent steps.

2) Configure the Java Messaging Service (JMS) queue destinations
Copy the file *rembrandt-jbossmq-destinations-service.xml* from the conf directory and place it in the <JBOSS_HOME>/server/default/deploy/jms/ directory on the machine where Rembrandt is to be deployed (this may or may not be the same machine running the analysis server).
Restart JBoss.

Two JMS queue destinations named AnalysisRequest and AnalysisResponse will now be available.

3) Configure the Analysis Server Build
Open the file basic-build.properties file in a text editor.
Change the property: **jms.location** to point to the machine hosting the Rembrandt application. The port number is the JBoss jnpPort number. For example, if the JBoss instance hosting Rembrandt is located on a machine called: rembrandt-server.myinstitution.org and jnpPort=1099 then the **jms.location** property would look like:

jms.location=rembrandt-server.myinstitution.org:1099

The jnpPort number can be found by looking at the JBoss startup output line containing the text **"INFO [NamingService]".**

Change the property: **analysis-server.install.dir** to contain the full path to the analysis server install directory. For example, if the *caintegrator-analysis-server.zip* file was downloaded and extracted into the directory /usr/applications then the **analysis-server.install.dir** property would look like:
analysis-server.install.dir=/usr/applications/caintegrator-ver1.0/caintegrator-analysis-server

Download the latest client jar file (use the exceptions version) from: http://stats.math.uni-augsburg.de/Rserve/down.shtml after agreeing to their licensing terms and conditions. The latest file at this time is: *JRclient-RE817.jar*

Copy the downloaded file into the lib directory under the analysis server install directory and rename it to *Rserve.jar* .

4) Build the Analysis Server

Change to the analysis server install directory and type **ant**. When the build completes successfully, the analysis server will be ready to run.

Running the Analysis Server

1) Go to the bin subdirectory of the analysis server install directory.

2) Enable permission to execute the script files using the Unix command **chmod**. For example, to give the owner of the files permission to execute the scripts type:
chmod u+x *.sh

3) Execute the Unix command: **dos2unix *.sh**

4) Make sure that the JAVA_HOME environment variable is set to point to the directory where the Java SDK is installed. For example if the Java SDK is installed in the directory /usr/j2sdk1.5.0_05 then issuing the command:
echo $JAVA_HOME should give output: /usr/j2sdk1.5.0_05

The Unix **export** or **set** command (depending on your shell) can be used to set the JAVA_HOME environment variable.

5) Start Rserve by typing: **./startRserve.sh**

Rserve should now be running. The last line printed on the console should be: "**Rserv started in daemon mode**."

6) Start the Analysis Server by typing: `./runAnalysisServer.sh`

You should see the output: **"nohup: appending output to `nohup.out'"**
(Hit the enter key to get to the Unix prompt)

The Analysis Server should now be running and listening for requests. To verify that it is running, change to the log directory and type the command: `tail analysisServer.log`. You should see a line with: **"Now listening for requests..."**. If instead, you see a line with: **"could not establish connection with provider.."** see the troubleshooting section below.

7) The analysis server will run continuously listening for and responding to requests. If the JMS provider (JBossMQ) goes down (for example if JBoss is restarted) it will automatically reconnect when JBossMQ becomes available.

To stop the analysis server, find the process ids associated with the analysis server and Rserve using the Unix `ps` command. The analysis server command name will start with java and the Rserve processes will start with Rserve. Then `kill` the processes using the Unix `kill` command.

Troubleshooting

- "**could not establish connection with provider..**" when starting the Analysis Server.

This error can occur when the Java Messaging Service (JMS) provider is not running or has not fully initialized. Make sure that the JMS provider that is configured in the analysis server build file is running. If it is running but has not fully initialized the analysis server will automatically connect when it is fully initialized.

- Analysis Server appears to be running but no results are returned.

Using the jmx-console (see the JBoss documentation) application go to the jboss.mq.destination section and inspect the AnalysisResponseQueue. Make sure that the receivers count equals 1. If it is greater than 1 then there is another listener on the queue which is stealing the results from application.

Next inspect the AnalysisRequestQueue. Make sure that the receivers count is equal to the number of analysis server machines. For example if you have configured four analysis server machines then the receivers count should equal four.
If the receiver counts are not as expected then the problem may stem from network/firewall issues or from a conflicting instance of the application.

- If you are using a version of JBoss other than 4.0.2, you should replace the following jar files:

concurrent.jar
jbossmq-client.jar
jboss-common-client.jar
jboss-system-client.jar
jnp-client.jar

in the lib directory under the analysis server install directory, with the corresponding jar files from the JBoss distribution.

Section 4

Rembrandt Application

Assumptions and Requirements

All previous components listed in this guide must be installed, configured, and deployed prior to installing this component.

In addition to the general requirements, Rembrandt utilizes the java.awt package, thus it may be required to start the java process with the following command when running on Unix and Linux:

```
"-Djava.awt.headless=true"
```

In a windows environment, this is not necessary.

There is currently a 2GB RAM minimum requirement for any server that this application is deployed on. This is due to the extremely large data sets that are being processed and returned to the user.

The following components are required and come as part of the bundled download.

Application Software: Distributed as part of Rembrandt 1.0 bundled download

Software	Version	Included in the bundled caIntegrator 1.0 D/L
webGenomeIntegration(WGI)	1.0	Yes
caintegrator-analysis-commons	1.0	Yes
caintegrator-spec	1.0	Yes
Rembrandt	1.0	Yes
WebGenome	2.0	No. Download available via the NCICB Download Center.
weka.jar	3-4-7	No

Download a zip archive containing the Weka source files and instructions to build the *weka.jar* from:

http://www.cs.waikato.ac.nz/~ml/weka/index.html after agreeing to their licensing terms and conditions.

The latest file at this time is: *weka-3-4-7.zip*

Unzip the archive. This will create a new directory called weka-3-4-7.

Copy the *weka.jar* file from the extracted zip into the /webRoot/WEB-INF/lib directory under the rembrandt install directory.

Installation/Deployment

Locate the *rembrandt.zip* file which is distributed as part of the caIntegrator 1.0 bundled download.

The Rembrandt web application is dependent on the following jars, as mentioned above: *caintegrator-analysis-commons.jar* and *caintegrator-spec.jar*. Both jars are distributed with the bundled download and can be found at : /rembrandt/WebRoot/WEB-INF/lib/. They can also be built from source using the build files bundled in *caintegrator-analysis-commons.zip* and *caintegrator-spec.zip* with the default target using ant if desired.

Within the downloaded bundle, locate the *build.properties* file located at the root of the extracted /rembrandt/ . Verify the following properties in the *build.properties* file, variables noted in < > notations should be replaced with your system settings (many of which are determined in previous sections of this guide):

Note: These properties refer to the location (JBoss instance) in which you will be deploying the Rembrandt application.

jboss.install.dir=<JBOSS_HOME>

jboss.server.name=default
(this variable refers to your JBoss instance. This guide assumes that you are using "default" as the JBoss instance name, modify if necessary.)

logging.dir=/<JBOSS_HOME>/server/default/log

analytical.jboss.jndi=<JMS_LOCATION>:<JMS_PORT>

(this pertains to the Analysis Server configuration and typically this is “localhost:1099” assuming you are using the built-in JBoss MQ for JMS, adjust as necessary)

urlDbalias=<DB_ALIAS>

databaseUser=<DB_USER>

databasePassword=<DB_PASSWORD>

webGenome.url= <WEB_GENOME_URL>/webGenome/client/plot.do
(if you are running rembrandt.war, webGenome.war and webGenomeIntegrator service (See Section 5 for bioAssayService.jar) on the same server, <WEB_GENOME_URL> can be “localhost”, else it will be the server name or IP of the machine Web Genome web application is running on. Web Genome is available for download separately via the NCICB download center as mentioned above.)

webGenomeJndi.url= jnp://<JNDI_URL>:<JNDI_PORT>
(this needs to be the same server name or IP as the webGenome web application, this can be “localhost:1099” assuming you are running the webGenomeIntegrator and the Web Genome web application on the same JBoss instance, adjust as necessary)

csm.hibernate.config.file=/<JBOSS_HOME>/server/default/conf/rembrandt.hibernate.cfg.xml
(this file will be created when you run the ant build, described below. Until then there will be no file located at the specified path)

csm.application.config.file=/<JBOSS_HOME>/server/default/conf/ApplicationSecurityConfig.xml
(this file will be created when you run the ant build, described below. Until then there will be no file located at the specified path)

Note: The Rembrandt application uses LDAP to manage the usernames and passwords. Without any further customization, you may use the username: RBTuser and password: RBTpass to bypass the LDAP authentication and access the application once installed. If you would like to use LDAP, please modify the 5 LDAP related properties, labeled “LDAP Properties”, within the *build.properties* file in accordance with your LDAP server settings as well. You will also need to complete the following section in its entirety. If you do not wish to use LDAP, you will need to modify the codebase to allow for authentication without LDAP.

Once your properties are all properly set, execute the ant build file’s (*build.xml*) default task which will create the *rembrandt.war* and associated files.

The Rembrandt web application relies on the NCICB CSM (Common Security Module) module for authorization and authentication. To setup the CSM component, please adhere to the following instructions:

1) Go to the newly created "csm_deploy" folder (which was created by running the ant build mentioned above and is located at /caIntegrator-ver1.0/rembrandt/csm_deploy/) and you should see the following files:

ApplicationSecurityConfig.xml
oracle-ds.xml
properties-service.xml
rembrandt.hibernate.cfg.xml

2) Copy the following files from the "/caIntegrator-ver1.0/rembrandt/csm_deploy" folder and place them in: /<JBOSS_HOME>/server/default/conf/

ApplicationSecurityConfig.xml
rembrandt.hibernate.cfg.xml

3) Copy the *oracle-ds.xml* from the "/caIntegrator-ver1.0/rembrandt/csm_deploy" folder and place it in: /<JBOSS_HOME>/server/default/deploy/

4) Copy the text that is inside the file *properties-service.xml*, located at "/caIntegrator-ver1.0/rembrandt/csm_deploy" which should look like:

```
<attribute name="Properties">
gov.nih.nci.security.configFile=/<JBOSS_HOME>/server/default/conf/Appli
cationSecurityConfig.xml
</attribute>
```

and add it to the

```
<mbean code="org.jboss.varia.property.SystemPropertiesService"
```

... tag inside the *properties-services.xml* file that you can find located in /<JBOSS_HOME>/server/default/deploy/

5) Optional: Complete only if you are using LDAP for authentication.

Add the following lines to
/<JBOSS_HOME>/server/default/conf/*login-config.xml*
Replacing the variables <LDAP_SERVER>, <LDAP_PORT>, <LDAP_OU_GROUP>, and <LDAP_O_GROUP> with your LDAP settings.

```
<application-policy name="rembrandt">
      <authentication>
            <login-module
code="gov.nih.nci.security.authentication.loginmodules.LDAPLoginModule"
flag="required">
            <module-option
name="ldapHost">ldaps://<LDAP_SERVER>:<LDAP_PORT></module-option>
            <module-option
name="ldapSearchableBase">ou=<LDAP_OU_GROUP>,o=<LDAP_O_GROUP></module-
option>
            <module-option name="ldapUserIdLabel">cn</module-option>
            </login-module>
      </authentication>
</application-policy>
```

6) Lastly, until the CSM is upgraded to use Hibernate 3.0 you must delete the conflicting folder located at: /<JBOSS_HOME>/server/default/deploy/jboss-hibernate.deployer.

This is due to a conflict between the different versions of hibernate.

7) Copy the file "*rembrandt-jbossmq-destinations-service.xml*" in the /caIntegrator-ver1.0/rembrandt/conf directory and place it in the /<JBOSS_HOME>/server/default/deploy/jms/ directory where Rembrandt is to be deployed.

8) Deploy the *rembrandt.war* file, located at /caIntegrator-ver1.0/rembrandt/, to the proper location /<JBOSS_HOME>/server/default/deploy/

Restart the JBoss container that owns the directory where this file was placed.

Verification

1) Use the following account to log in for testing of the application:
Username: RBTuser
Password: RBTpass
Start JBOSS and navigate your web browser to:
http://<JBOSS_SERVER:PORT>/rembrandt/
(for example http://localhost:8080/rembrandt)

2) You should be greeted with the legal rules of the road page, press "clicking here" and you should go to the next page, which is the "Simple Search" Tab

3) Select the option for "Kaplan-Meier survival plot for Gene Expression Data" and type "EGFR" in the text field next to "Gene Keyword". Press the "Go" button and this should take you to a page that shows a Kaplan-Meier graphical plot.

4) Click on the “Higher Order Analysis” tab which should take you to a page with three buttons:

“Class Comparison”
“Principal Component Analysis (PCA)”
“Hierarchical Clustering Analysis”

5) Press the “Principal Component Analysis (PCA)” button which should take you to a new page that allows you to construct a Higher Order Analysis request.

6) Name the “Analysis Result” something and press “submit”. This should take you to the “View Results” page where you should see the Analysis request you submitted with a “running…” message. After some unknown time you should see this change to a check mark with a “completed” message.

7) Click the link that is the name that you gave the Analysis Result in Step 6 and you should see a pop up of a PCA graph with 3 tabs.

Known Issues

- java.io.IOException: Stream closed
  This exception may occur on the JBoss console during normal application usage. This is a known issue and will be addressed in a future release.

Section 5

WebGenome Integration

webGenomeIntegration (WGI) module provides BioAssay Data Service for caIntegrator repository. It serves various BioAssay quantitation types such as Copy Number, SNP, Gene Expression etc. The current release supports Copy Number data only. It utilizes EJB technology (Stateless Session beans) for exposing the services. WGI also utilizes caIntegrator multi-threaded query service for retrieving the bioassay data from caIntegrator repository.

webGenome (formerly webCGH) is a web-base application that provides genomic data plotting capabilities. An interface between caIntegrator and webGenome was created to leverage the plotting capabilities of webGenome for data store in the caIntegrator datawarehouse. WebGenome exposes a set of interfaces to help populate these plots and the caIntegrator framework implements these interfaces as Enterprise Java Beans (EJB). caIntegrator remotely invokes the WebGenome application via the Java Naming and Directory Interface (JNDI).

WGI is purely a service-tier component and does not have any UI component associated with it. It is consumed by both the Rembrandt Application and WebGenome applications. To facilitate the application integration, an Application State service was also added to WGI. Please refer to technical design document (Rembrandt 1.0) for more details on these services.

Assumptions and Requirements

In addition to meeting the general requirements, the following are also required:

Application Software:

Software	Version	Included in the bundled caIntegrator 1.0 D/L
webGenomeIntegration(WGI)	1.0	Yes

caintegrator-analysis-commons	1.0	Yes
caintegrator-spec	1.0	Yes
Rembrandt	1.0	Yes
WebGenome	2.0	No. Download available via the NCICB Download Center.

Third Party Software: Comes as part of Rembrandt 1.0 download

Software	Name/ Version	Included in the bundled caIntegrator 1.0 D/L
Object Relational Mapping (ORM)	Object Java Bridge (OJB) 1.0	Yes
JDBC Drivers	Oracle JDBC Thin drivers	Yes

Please ensure both *Rembrandt.zip* and *webGenenomeIntegration.zip* have been extracted to the /caIntegrator-ver1.0 folder.

Note: WGI and the WebGenome web application need to be deployed on the same server. For better performance, it is recommended that you install WGI and WebGenome on a separate server than where the Rembrandt web application is installed.

If you do plan to install WGI, WebGenome Application and Rembrandt Application on the same JBOSS instance then the following settings needs to be modified in the JBOSS configuration.

1. Modify <JBOSS_HOME>/server/default/deploy/ear-deployer.xml file as follows:
   change: "<attribute name="CallByValue">**false**</attribute>"
   to: "<attribute name="CallByValue">**true**</attribute>"

2. Verify that within <JBOSS_HOME>/server/default/deploy/jbossweb-tomcat55.sar/META-INF/jboss-service.xml file, the status for the following attributes is set to false and modify to false if necessary:

<attribute name="UseJBossWebLoader">**false**</attribute>

<attribute name="Java2ClassLoadingCompliance">**false**</attribute>

Installation

Note: The home directory of your JBoss default installation (for example: c:\JBoss-4.0.2\) will be referred to as <JBOSS_HOME> throughout this section.

The build script (and associated *build.properties*) to build the WGI module can be found under the /caIntegrator-ver1.0/webGenomeIntegration directory. The default tag of this build script will create a jar file called "*bioAssayService.jar*". The deployment of WGI simply consists of dropping this jar file into the "deploy" directory of your JBoss installation (such as <JBOSS_HOME>\server\default\deploy).

Note: As mentioned in the Assumptions section, building the WGI module depends on 3 other projects: *caintegrator-analysis-common.zip*, *caintegrator-spec.zip,*and *rembrandt.zip*. By running the WGI build file, we will generate the necessary jar files from the source of each of the 3 dependencies. Follow the steps below to complete the build:

1) Configure below properties in *build.properties* file under the /caIntegrator-ver1.0/webGenomeIntegration directory.

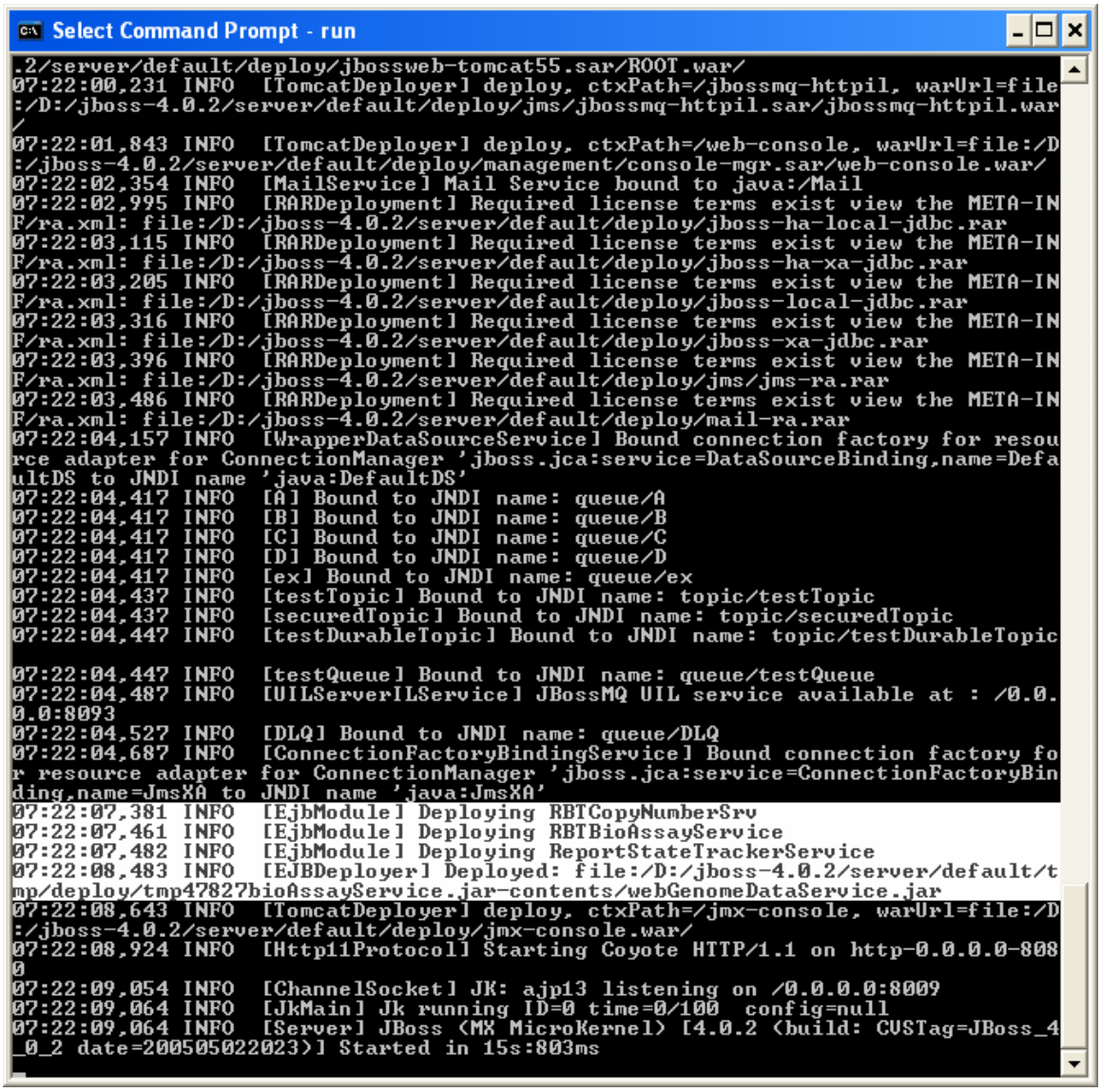
   i. **webGenomeJndi.url**: This should point to the <JNDI_URL>:<JNDI_PORT> where webGenomeIntegration will be deployed.
   NOTE: This needs to be the same server name or IP as the webGenome web application, this can be "localhost:1099" assuming you are running the webGenomeIntegrator and Web Genome web application on the same JBoss instance, adjust as necessary)
   ii. **urlDbalias**: This should point to the JDBC URL. For example: "thin:@xyz.org:port:INSTANCE"
   iii. **databaseUser**: This should have your Database-username
   iv. **databasePassword**: This should have your Database-password

2) run "**ant**" command from /caIntegrator-ver1.0/webGenomeIntegration directory. If every property is properly configured, then you will see the generated "*bioAssayService.jar*" file under the *checkoutDir* directory.

3) Copy the newly generated "*bioAssayService.jar*" to the deploy directory of JBoss's default installation (for example: <JBOSS_HOME>\server\default\deploy)

4) Configure WGI logging using Log4j. This step involves copying the log4j appenders and categories from WGIRoot /src/log4j-bioAssayService-appenders.xml in to jboss's *log4j.xml* file under <JBOSS_HOME>/server/default/conf directory.

5) Test the deployment of WGI, start JBoss and you should see the below highlighted messages in the console.



Troubleshooting

- If WGI is properly deployed, then you should see the above highlighted messages in the console. If your installation was not successful, errors will appear on the console.
- WGI logs all EJB requests for both ApplicationStateTracker and BioAssayService using Log4j. The log file, *bioAssayService.log*, can be found under the log folder of the corresponding jboss container where the WGI is deployed (e.g. <JBOSS_HOME>/server/default/log). This log file includes the SQL statements that are executed in a multi-threaded fashion.

Verification

1) Make sure Rembrandt war file has been properly build and deployed and the following properties were properly configured during the build (should be the same for both Rembrandt and WGI module)
webGenome.url= <WEB_GENOME_URL>/webGenome/client/plot.do

webGenomeJndi.url= jnp://<JNDI_URL>:JNDI_PORT

2) Use the following account to log in for testing of the application:
Username: RBTuser
Password: RBTpass
Start JBOSS and navigate your web browser to
http://<JBOSS_SERVER:PORT>/rembrandt/

3) Go to Advance Search , Select copy number
   a. Type query name "WGI Test"
   b. Type "EGFR" as gene name
   c. Select "Astrocytomoa" as Disease Type
   d. Select Preview, A preview Report should pop up.
4) Click on "View Graph (Via webGenome)".
5) A new WebGenome window will popup with graph.

Deployment Recommendations

For best performance it is recommended that WGI be deployed in the same container wherever the WebGenome application is deployed. Derived applications, such as Rembrandt, should be deployed on its own application server.

WGI Services

WGI mainly provides two types of services.
1) ApplicationStateTracker
This is consumed by the Rembrandt application. The corresponding EJB client jar file, **"***appStateClient.jar***"**, is included with the Rembrandt source distribution.

2) BioAssayService
This is consumed by the WebGenome application. The corresponding EJB client jar file, **"***webGenomeClientEJB.jar***",** is included with the WebGenome source distribution.

Note: Both of these jars can also be created from the WGI source using the corresponding ant tags **"**buildAppStateClient**"** and **"**buildWebGenomeClient**"** located in the WGI build script.

Section X

Appendix A: caIntegrator-spec

caIntegrator-spec is the module that contains the source code for the caintegrator's generic application framework upon which various NCICB and caBIG translational research applications such as Rembrandt and I-SPY are being built. Applications built upon caIntegrator-spec will allow researchers to access and analyze clinical and experimental data collected across multiple participating institutes and study time points.

Note: the bundled distribution includes the pre-built *caintegrator-spec.jar*, please use the information in this section only if you intend to modify the source for this library and recompile it yourself.

caIntegrator-spec contains the following:

Package	Description
gov.nih.nci.caintegrator.domain	Clinical Genomic Object Model objects
gov.nih.nci.caintegrator.dto	Data Transfer Objects
gov.nih.nci.caintegrator.service.findings	FindingsFactory & Findings Interfaces
gov.nih.nci.caintegrator.service.findings.strategies	FindingStrategy Interface
gov.nih.nci.caintegrator.ui.graphing	Graphing Objects for Kaplan Meier, Clinical and PCA plots
gov.nih.nci.caintegrator.security	CSM enabled Security components

Assumptions and Requirements

In addition to the general requirements being met, the following assumptions are in place: You have properly downloaded the caintegrator bundle, and located the nested archive, *caintegrator-spec.zip*, after extracting the bundle.

Application Software:

Software	Version	Included in the bundled caIntegrator 1.0 D/L
caintegrator-analysis-commons	1.0	Yes
weka.jar	3-4-7	No

Download a zip archive containing the Weka source files and instructions to build the *weka.jar* from:

http://www.cs.waikato.ac.nz/~ml/weka/index.html after agreeing to their licensing terms and conditions.

The latest file at this time is: *weka-3-4-7.zip*

Unzip the archive. This will create a new directory called weka-3-4-7. To run Weka, navigate to that directory and type:

```
java -jar weka.jar
```

Copy the downloaded file into the /deployed_jars directory under the caintegrator-spec install directory.

Initialization

Follow the sequence to properly recompile the caIntegrator-spec source code or download the caIntegrator-spec.jar binary file.

1) Accessing the file *caintegrator-spec.zip*

On the machine(s) that will be used to run the Rembrandt web application, extract the file *caintegrator-spec.zip* to the directory where the software is to be installed.
The subdirectory named caintegrator-spec which is located under the caintegrator-ver1.0 directory will be referred to as the "caIntegrator-spec directory" in subsequent steps.

2) Compiling the caIntegrator-spec source code

Run the default ant command from the caIntegrator-spec directory to recompile and jar the caIntegrator-spec source code or use the bundled *caIntegrator-spec.jar* file.

Appendix B: caIntegrator-analysis-commons

caIntegrator-analysis-commons module contains all the dependency objects such as analysis messages, enums and exceptions that are necessary to compile the caIntegrator-spec, caIntegrator-analysis-server and Rembrandt web application modules.

Note: the bundled distribution includes the pre-built *caintegrator-analysis-commons.jar*, please use the information in this section only if you intend to modify the source for this library and recompile it yourself.

CaIntegrator-analysis-commons contains the following:

Package	Description
gov.nih.nci.caintegrator.analysis.messaging	Analysis messaging objects utilized by both the Analysis Server module and the Rembrandt application
gov.nih.nci.caintegrator.enumeration	Various enum objects used by various caIntegrator classes
gov.nih.nci.caintegrator.exceptions	Exception Objects used by all other caIntegrator modules

Assumptions and Requirements

In addition to the general requirements being met, the following assumptions are in place: You have properly downloaded the caintegrator bundle, and located the nested archive, *caintegrator-analysis-commons.zip*, after extracting the bundle.

Initialization

Follow the sequence to properly recompile the caintegrator-analysis-commons source code or download the caIntegrator-analysis-commons binary file.

1) Accessing the file *caIntegrator-analysis-commons.zip*

On the machine(s) that will be used to run the Rembrandt web application, extract the file *caIntegrator-analysis-commons.zip* to the directory where the software is to be installed. The subdirectory named caintegrator-analysis-commons which is located under the caintegrator-ver1.0 directory will be referred to as the "caIntegrator-analysis-commons directory" in subsequent steps.

2) Compiling the caintegrator-analysis-commons source code

Run the default ant command from the caintegrator-analysis-commons directory to recompile and jar the caintegrator-analysis-commons source code or use the bundled *caintegrator-analysis-commons.jar* file.

Appendix C : caIntegrator Downloads

The following appendix describes what is included in the caIntegrator distribution and how to obtain the relevant files. All files are available via the NCICB Download Center or via the caIntegrator website: http://caintegrator.nci.nih.gov/caintegrator/downloads/

Documentation

The following documentation provides instructions for downloading, installing, and using the caIntegrator and its reference implementation. See the release notes for the latest features and updates, and the Java Docs (caIntegrator + REMBRANDT) for additional details.

Installation Instructions: *caIntegrator_Install_Instructions.pdf*
Release Notes: *caIntegrator1.0_releaseNotes.txt*
User Guide: *rembrandt1.0_ug_12_20_05_nsn.pdf*
caIntegrator Bundle/Javadocs: *caIntegrator1.0_javadocs.zip*

Source Distribution

The source code distribution below contains the caIntegrator source and its reference implementation (REMBRANDT).

Source Code: *caIntegrator1.0_src_bundle.zip*

Database Download

The file below contains all necessary instructions and information for importing the database file for the REMBRANDT reference implementation .

Database Instructions: *rembrandt_db_instructions.zip*

Appendix D: 3rd Party Tools

3rd Party Tool	Version	URL	License Info/URL
ehCache	1.1	http://ehcache.sourceforge.net/	http://ehcache.sourceforge.net/ Available under the Apache 1.1license. Ehcache's copyright and licensing has been reviewed and approved by the Apache Software Foundation, making ehcache suitable for use in Apache projects.
JFreechart	1.0.0 rc1	http://www.jfree.org/jfreechart/index.php	http://www.jfree.org/lgpl.php complete source code is included, under the terms of the GNU Lesser General Public Licence; http://www.jfree.org/jfreechart/index.php
Apache Struts	1.1	http://struts.apache.org/	
Rserve		http://stats.math.uni-augsburg.de/Rserve/	GPL software **license**
DWR	1.0	http://getahead.ltd.uk/home	http://www.apache.org/licenses/LICENSE-2.0.html
DOM4J	1.5.2	http://www.dom4j.org/	http://www.dom4j.org/license.html BSD License
Hibernate	2.1.7c	http://www.hibernate.org	http://www.hibernate.org/356.html BSD License

Krysalis-jCharts	1.0.0-alpha-1	http://jcharts.sourceforge.net/index.html	http://jcharts.sourceforge.net/license.html BSD License
log4j	1.2.8	http://logging.apache.org/log4j/docs/	http://vmgump.apache.org/gump/public-jars/nlog4j/jars/LICENSE.txt
p6spy		http://www.p6spy.com/	http://www.ibiblio.org/maven/p6spy/licenses/license.htm
velocity	1.3.1	http://jakarta.apache.org/velocity/	http://jakarta.apache.org/velocity/docs/license.html
xdoclet	1.2b3-dev	http://xdoclet.sourceforge.net/	BSD http://xdoclet.sourceforge.net/xdoclet/licenses/xdoclet-license.html
xjavadoc	1.0	http://xdoclet.sourceforge.net/xjavadoc/	
db-ojb-1.0.rc4-src.jar	1.0.rc4		
jakarta regexp	1.2	http://jakarta.apache.org/regexp/	
Java API for Servlets - javax.servlet.jar javax.servlet.jsp.jar			
JBOSS -j2ee		http://www.jboss.org/	
Java Data Objects Sun's reference implementation of the JDO specification	jdo.jar jdori.jar	http://www.javaranch.com/newsletter/200401/IntroToJDO.html	
antlr.jar		http://www.antlr.org/	http://www.antlr.org/license.html
appStateClient.jar			
batik-awt-util.jar		http://xml.apache.org/batik/	
batik-dom.jar		http://xml.apache.org/batik/	
batik-svggen.jar		http://xml.apache.org/batik/	
batik-util.jar		http://xml.apache.org/batik/	
batik-xml.jar		http://xml.apache.org/batik/	
c3p0-0.8.4.5.jar			
caBIO	caBIO.jar	http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/caBIO	
cglib-full-2.0.2.jar	2.0.2		
commons-beanutils.jar		http://jakarta.apache.org/commons/beanutils/	
commons-collections-3.1.jar	3.1	http://jakarta.apache.org/site/downloads/downloads_commons-collections.cgi	
commons-dbcp.jar			
commons-digester.jar		http://jakarta.apache.org/commons/digester/	
commons-fileupload.jar			
commons-lang-2.0.jar			
commons-logging	1.0.4.jar	http://jakarta.apache.org/commons/logging/	
commons-pool.jar			
commons-validator.jar			
csmapi.jar		http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/csm/	
jaas.jar		http://java.sun.com/products/jaas/install_notes.html	
jakarta-oro.jar		http://jakarta.apache.org/oro/	

jcommon-1.0.0-rc1.jar	1.0.0
jcs.jar	
jdo.jar	
odmg-3.0.jar	3.0
ojdbc14.jar	